KNIGHTINK

SCORCHED EARTH:
HACKING BANKS AND CRYPTOCURRENCY EXCHANGES THROUGH THEIR APIS

APIs are the plumbing of today's financial services and FinTech infrastructure, enabling FinTechs to embed banking into their apps and banks to offer a more unified experience to their customers demanding more from their bank. However, when these APIs are vulnerable to attack and the vulnerable code is reused in hundreds of other banks by an outsourced developer, it can become systemic, creating an attack surface pervasive across our nation's financial system. This white paper presents a year of research into the exploration of those vulnerabilities and what can happen when they're discovered.

SUMMARY

This paper unveils a year of vulnerability research into hacking banks, neobanks, and cryptocurrency apps and APIs. The findings in this report affect 55 financial services and FinTech mobile apps and APIs and represents the largest unveiling of vulnerabilities in the financial services and FinTech industries in history.

AUTHOR INFORMATION

Alissa Valentina Knight
Partner
Knight Ink
1980 Festival Plaza Drive
Suite 300
Las Vegas, NV 89135
ak@knightinkmedia.com

PUBLICATION INFORMATION

This white paper is sponsored by Noname Security.

Initial Date of Publication:
DECEMBER 2021
Revision: 1.0

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

KEY TAKEAWAYS

- APIs are the plumbing of today's financial services and FinTech infrastructure, enabling FinTechs to embed banking into their apps and banks to offer a more unified experience to their customers demanding more from their bank.
- 54 of the 55 mobile apps that were reverse engineered contained hardcoded API keys and tokens including usernames and passwords to third-party services.
- All 55 apps tested were vulnerable to woman-in-the-middle (WITM) attacks, allowing Knight to intercept and decrypt the encrypted traffic between the mobile apps and backend APIs
- One of the banks outsourced development of their mobile app and APIs. That developer re-used the same vulnerable code affecting 300 of its other bank customers.
- 100% of the APIs tested were vulnerable to OWASP API1:2019 Broken Object Level Authorization (BOLA) vulnerabilities allowing Knight to change the PIN code of any bank customer's Visa ATM debit card number and transfer money in and out of accounts.
- Due to a failure to authenticate the API requests, the APIs tested were also vulnerable to OWASP API2:2019 Broken Authentication, which allowed me to transfer money in and out of different bank accounts and change customer ATM debit PIN codes as long as I knew the account numbers without authentication.
- The APIs tested were vulnerable to Broken Authentication vulnerabilities allowing Knight to perform API requests on other bank customer accounts without authenticating.
- APIs were deployed behind web application firewalls (WAFs) -- the wrong security control, incapable of detecting logic-based attacks like authentication and authorization vulnerabilities
- During several of the engagements, some of the banks were unable to find specific API endpoints affected by the vulnerabilities indicating a clear visibility problem into their API attack surface.

An Application Programming Interface (API) is an intermediary between software components allowing them to talk, one acting as an API consumer (API client) and the other as the API provider (API endpoint/s). Essentially, an API acts somewhat like a Rosetta Stone that allows different applications to talk to one another so data can be shared between them. APIs also act somewhat like an agreement that defines a contract between applications that the API endpoint will receive specific requests and in exchange, will respond back with specific responses.

INTRODUCTION

INTRODUCTION

Before getting into the tactics and techniques in hacking financial services and financial technology (FinTech) APIs, it's important to first define what APIs are and how they're transforming traditional banking as we once knew it.

This paper is structured first by presenting front matter on a short history and background on APIs and then how APIs are transforming the banking and FinTech sector. Next, I present on the tactics and techniques I used in this research to hack the 55 banking and FinTech mobile apps and APIs that were targets of evaluation (TOE).

WHAT ARE APIs

An Application Programming Interface (API) is an intermediary between software components allowing them to talk, one acting as an API consumer (API client) and the other as the API provider (API endpoint/s). Essentially, an API acts somewhat like a Rosetta Stone that allows different applications to talk to one another so data can be shared between them. APIs also act somewhat like an agreement that defines a contract between applications that the API endpoint will receive specific requests and in exchange, will respond back with specific responses.

Prior to the advent of APIs, meaning before February 7, 2000 when SalesForce first debuted the world's first XML API at the IDG Demo 2000 conference, web applications were monolithic, a software design architecture where the application is written in a single coherent piece of code, often on a single server.

Enter Microservices where the software design architecture is that of an application which has been broken down into separate, interconnected components, each with its own logic and its own processes that communicate with one another using APIs.

APIs IN FINANCIAL SERVICES

APIs are completely transforming traditional banking as we once knew it; enabling traditional brick and mortar banks to digitize their services and offer products to bankable customers who don't live near physical branches or service the new generation of consumers wanting an internet-bank. Branch and ATM banking have been in steady decline among consumers for years, while mobile banking has skyrocketed. Mobile banking was the primary banking channel for 34 percent middle to high income people under age 54 in 2019, compared to about 6 percent in 2013, according to FDIC data cited by ABA Banking Journal.

APIs also enable banks and FinTechs to work together in a unified ecosystem to deliver a more frictionless experience to customers demanding more from their bank. Today's consumer wants all of the places their financial information resides across different platforms to be able to talk to each other and share data. APIs allow banks to provide digital services to their customers as well as integrate with other digital services provided by FinTechs ultimately to enable them to embed banking services into their applications.

The ecosystem of FinTechs is expanding and today includes apps, such as neobanks (digital banks with no brick-and-mortar branches), cryptocurrency exchanges, person-to-person (P2P) payment apps, and more.

The legacy banking system is undergoing a rapid modernization — creating an increasingly connected banking infrastructure to support things like open banking (PSD2 in the UK) and connectivity between banks and FinTechs are APIs.

PURPOSE

The purpose of this report is to provide irrefutable evidence through empirical data collected through a vulnerability research campaign into hacking financial services and FinTech APIs over the period of a year as to what can happen when vulnerabilities exist in the APIs of our financial system.

This research, referred to as adversarial content, makes an argument for why API security must quickly move to a required line item in the Chief Information Security Officer's (CISOs) budget as a compulsory security control as a matter of necessity. The APIs tested in this research were either not protected by an API threat management solution or the wrong security control was used to protect them, such as a web application firewall (WAF) or API gateway.

Adversarial content attempts to prove the need and efficacy of a security control by hacking the very targets it aims to protect. This research, sponsored by Noname Security, a leader in API threat management whose solution effectively prevents the tactics and techniques used in this study.

RESEARCH METHODOLOGY

The type of research performed for this report was primary research. A total of 55 financial services and FinTech mobile apps covering 19 banks, 11 cryptocurrency exchanges, and 21 neobank apps were covered targets.

The first step in the research performed was to select the targets under evaluation for each category of banking and FinTech apps, then downloading them from the Google Play app store. The apps were extracted off of the Android device and placed on my analysis workstation for reverse engineering where they were reversed back to their original source code where static code analysis was performed against the apps to find hardcoded API secrets (keys and tokens) in each app.

The apps were then installed on an Apple iPhone where traffic interdiction was performed of each app to test for the presence or absence of certificate pinning allowing for a woman-in-the-middle (WITM) attack to be employed against the app. This was used to footprint the backend API endpoints to take the attacks further into an API client where manual API requests could be sent to them in order to test them outside their legitimate mobile app against the OWASP API Security Top 10, such as Broken Object Level Authorization (BOLA) and Broken Authentication attacks.

API SECURITY MARKET

API SECURITY MARKET

Market Trends	**Market Implications**
Web Application Firewalls are attempting to remain relevant in today's microservices/API-first world by going after the API security budget.	CISOs are relying on WAFs to secure their APIs, which creates a false sense of security. Some of the APIs breached in this research study were deployed behind WAFs that failed to detect and prevent the logic-based attacks, such as Broken Object Level Authorization and Broken Authentication because they weren't designed to do so.
API threat management solutions can be categorized along the shift-left and shield-right horizontal line and further subdivided into where they sit in the network to detect threats. Some API threat management solutions sit inline acting as a sort-of API firewall, one option actually shims into the backend application for richer application context, and some analyze network traffic passively out-of-band relying on down-stream technologies to block attacks.	There is no right or wrong way to approach the architecture of where an API threat management solution sits in the network. It should be based on your individual risk appetite for inline solutions, proxying all of the traffic to your APIs or, in the case of shimming, whether or not it supports the language your API was developed in. Furthermore, no matter what, buyers should create a functional requirements document (FRD) to determine what your needs are as a must-have from your API threat management solution. This could be observability, API runtime security, and understanding which APIs serve what kind of data, and more.

Market Trends	Market Implications
The number of API endpoints that organizations are running has increased significantly over the past few years with organizations running an average of 1600 APIs now and many not running API threat management solutions to secure them.	The growing API attack surface has prompted many analysts, such as Gartner, to predict that next year, API breaches will be the number one attack vector used by adversaries. The adoption of an API threat management solution must no longer be an option and must move to the compulsory list of budget line items.
Branch and ATM banking have been in steady decline among consumers for years, while mobile banking has skyrocketed.	With the oldest Millennial recently turning 40 and gen-z becoming the new consumer, great demands are being placed on traditional brick and mortar banks towards digitalization and federation of their financial data spread across different FinTechs. This digitalization of newly connected apps is creating an attack surface that is becoming increasingly more vulnerable due to a rush to push out capabilities prompted by this increase in consumer demand. Security, as usual, is an afterthought and is introducing significant vulnerability into our banking system that wasn't previously anticipated.

THE RESEARCH

THE RESEARCH

From December of 2020 to December of 2021, I performed research into the vulnerability of traditional banking, neobanks, and FinTech mobile apps and APIs. The results of that research, while startling, are presented in this next section. Great care has been taken to obfuscate the identity of the targets under evaluation and participating banks and FinTech companies. Screenshots are presented in this section that have been purposely redacted by blurring identifying information.

TARGETS OF EVALUATION

The apps and APIs targeted in this research are categorized into their different demographics below. All banks, FinTechs, and and cryptocurrency exchanges have been de-identified to protect their anonymity in the research and thus the fundamentals and technical of the company have been presented here instead. Where some data was not able to be immediately found, the company was placed into the undisclosed category.

BANKS

The targets in this category are defined as traditional “Main Street” brick and mortar banks with branches. The banks in this category are not purely digital despite having mobile apps and offering digital services.

BANKS
The targets in this category are defined as traditional “Main Street” brick and mortar banks with branches. The banks in this category are not purely digital despite having mobile apps and offering digital services.

Headquarters	US	19
Number of Employees	1-10 K	2
	10 K - 50 K	8
	51 K - 100 K	5
	101 K+	4
Assets Under Management	$1 Mn - $1 Bn	0
	$1 Bn- $100 Bn	3
	$100 Bn - $500 Bn	11
	$500 Bn - $1 Tn	1
	$1 Tn+	4
Account Holders	Undisclosed	10
	1-500 K	0
	500 K - 1 Mn	0
	1 Mn - 10 Mn	3
	11 Mn - 50 Mn	3
	51 Mn - 100 Mn	2
	101 Mn+	1

NEOBANKS

The banks in this category are direct banks that offer their banking products exclusively over digital channels and do not operate traditional brick and mortar branches. These banks offer mobile apps and other online services are also referred to as challenger banks in the United Kingdom (UK).

Headquarters	US	17
	UK	3
	Canada	1
Number of Employees	1-50	5
	51 - 100	4
	101 - 500	7
	500 - 1000	0
	1001+	4
	Undisclosed	1
Assets Under Management	$1 Mn - $1 Bn	3
	$1.1 Bn - $5 Bn	2
	$5.1 Bn - $10 Bn	3
	$10.1 Bn - $100 Bn	0
	$100.1 Bn+	0
	Undisclosed	13
Account Holders	Undisclosed	7
	1-500 K	2
	501 K - 1 Mn	0
	1.1 Mn - 10 Mn	10
	10.1 Mn - 50 Mn	2
	50.1 Mn - 100 Mn	0
	100.1 Mn+	0

CRYPTOCURRENCY EXCHANGES
These companies operate currency exchanges online and offer no physical branches for customers. These exchanges allow customers to trade traditional currency, such as their country's fiat currency (such as the U.S. Dollar, Euro, British Pound, etc) for cryptocurrency and other forms of digital currency, such as Ethereum, Bitcoin, Cardano, and others, including but not limited to Non-Fungible Tokens (NFTs).

Headquarters	US	5
	UK	1
	Asia	3
	Other	2
Crypto Under Management	$1 Mn - $1 Bn	1
Account Holders	$1.1 Bn- $10 Bn	2
Assets Under Management	$10.1 Bn - $50 Bn	2
	$50.1 Bn+	1
	Undisclosed	5
Number of Employees	1 - 100	2
	101 - 250	4
	251 - 500	1
	500+	4

STATIC CODE ANALYSIS

STATIC CODE ANALYSIS

Static analysis, also referred to as a dead code analysis, is the “white box” review of an application’s source code for vulnerabilities prior to runtime — meaning, the code isn’t analyzed during execution, which is antithetical to static analysis would be dynamic code analysis.

When performing static analysis, the APK files for each mobile app were first extracted off of the Samsung Android phone I was using to install the apps. Using a freely available tool in the Google Play Store (APK Extractor), I extracted the apps off my mobile device and uploaded them to my Google Cloud Drive.

Once there, I downloaded the APK file to my Mac Pro 2020 where Mobile Security Framework (MobSF), which is an automated, all-in-one mobile application (Android/iOS/Windows) pen-testing, malware analysis and security assessment framework capable of performing static and dynamic analysis. MobSF supports mobile app binaries (APK, XAPK, IPA & APPX) along with zipped source code and provides REST APIs for integration with CI/CD or DevSecOps pipelines.

Hardcoded API secrets (keys and tokens) were discovered in 54 of the 55 apps tested (APPENDIX A). The specific results of the static code analysis for each category of apps is provided below.

Banks (30)

Lacks Obfuscation	27
Vulnerable to WITM	30
Hardcoded Keys	30

Neobanks (20)

Lacks Obfuscation	17
Vulnerable to WITM	15
Hardcoded Keys	17

Cryptocurrency Exchanges (11)

Lacks Obfuscation	10
Vulnerable to WITM	10
Hardcoded Keys	7

TOTALS

Lacks Obfuscation	54 / 55
Vulnerable to WITM	55 / 55
Hardcoded Keys	54 / 55

API Exploitation

API EXPLOITATION

ID: V001	**Vulnerability ID**	OWASP API2019:1
	Name	Broken Object Level Authorization
Description	API vulnerable to Broken Object Level Authorization allowing an adversary to change the PIN code for the ATM debit card and move money between accounts of any customer of the bank.	
Number of API Endpoints Vulnerable	300	
Evidence	APPENDIX B	

ID: V003	**Vulnerability ID**	OWASP API2019:2
	Name	Broken Authentication
Description	Due to a flaw in the code and absence of OAuth 2 tokens, it was possible to perform attacks V001 and V002 without authenticating.	
Number of API Endpoints Vulnerable	300	
Evidence	APPENDIX C	

SOLUTION

SOLUTION

API usage has surged into a sprawl for businesses of all shapes and sizes. Words and phrases like “digital transformation”, “cloud migration”, “apps”, and “microservices” all mean the same thing — lots and lots of APIs.

In my research, I found the same API security vulnerabilities in banks that had 25,000 customers and a few million in managed assets as I did in banks that had 68 million customers and $7.7 trillion in assets under management. Large, mature, and well-funded security teams are not able to keep pace with API security challenges with traditional tools and processes.

API Security needs to be operationalized across the enterprise. Many teams play critical roles at securing APIs. Developers need to write code with security in mind; cloud and platform teams need to use APIs that are configured properly; and security teams need to detect, investigate, and respond to incidents. Often, especially in larger organizations, APIs are deployed to production faster than they can be secured and there often isn’t a clear line of communication across enterprise teams.

Specific to my research, the APIs I exploited were developed by a third party — introducing yet another variable. Whatsmore is that the hack wasn’t detected at any of the banks. This highlights the fact that API security needs to be operationalized across more enterprises to ensure that vulnerabilities are detected and remediated before an attack. And it’s not just the responsibility of a single team. Developer, DevOps, DevSecOps, and security teams need to standardize, collaborate, and communicate how they build, deploy, and secure APIs.

API security requires posture management, runtime security, and active testing. It’s very easy to jump to conclusions when exploits or attacks make headlines. But detecting and blocking behavior like mine is only a piece of the API security puzzle. Enterprises need to think about API security across 3 core areas: (1) API Security Posture: organizations need a complete API inventory (including associated data and metadata) so they have a stronger sense of their security posture. It’s imperative to identify and remediate misconfigurations and vulnerabilities before an incident occurs. As evidenced in my research, many organizations are completely exposed and won’t be aware until after an attack has occured; (2) API Runtime Security: organizations need better visibility into the traffic and behavior of their APIs. This provides better detection and response to anomalous and suspicious behavior so attacks can be prevented in real-time when something out of the ordinary occurs (like I demonstrated here); and (3) API security testing: organizations need to identify security gaps as part of the software development lifecycle. For example, at no point should business critical APIs be deployed into production if they couldn’t pass basic security checks (e.g. lack of authentication and authorization). Active testing ensures confidence in your APIs throughout the lifecycle of an API.

25

CONCLUSION

CONCLUSION

Despite the availability of the OWASP Top 10 web application security guide, the OWASP API Security Top 10, and other secure coding guidance and training for developers, we still seem to be suffering from authentication and authorization issues systematically across different industries with financial services and FinTech included.

What was evident in my research was also the number of organizations whose IT teams were deploying APIs with WAFs offered by their cloud service provider without involving the CISOs office. There must be more cohesion and collegial work between the CTO and CISO's office when it comes to development operations (DevOps). Developers can't be deploying APIs into production without first having the cybersecurity team involved to ensure there is an effective level of security controls protecting the APIs being deployed and that they are being monitored appropriately moving forward post-production.

In the cases where APIs were being secured with the wrong controls (a la WAFs), I opine that it actually does more harm to the organization than not having any security controls in place at all. This is because if the wrong security control is being used to protect the APIs, this effectively creates a false sense of security versus an organization that knows they need to select and implement the appropriate API threat management solution for their business.

The systemic lack of certificate pinning in apps as sensitive as banking and cryptocurrency exchanges is also unfathomable. With consumers now banking from everywhere on their mobile phones, the likelihood of a successful WITM attack becomes more pervasive than it did 10-15 years ago. While I understand the threat of bricking an app due to certificate issues when pinning is implemented, the threats certainly outweigh the risks in today's API-first world.

API threat management solutions must become a compulsory purchase as part of the CISOs budget. Adversaries have a single goal in mind: to profit off the data your organization transacts, processes, and stores. As such, they are quickly shifting their attention and honing their craft to target APIs knowing that is where the data is sitting.

I'll end this paper a quote from Anonymous "If I were to advise a rogue nation state on how to take down the United States, I'd tell them to start with the APIs first."

BIBLIOGRAPHY

- The land before modern APIs. (2020, August 20). What the History of the HTTP Status Code Can Tell Us about the Future of API Design. Retrieved December 5, 2021, from https://increment.com/apis/land-before-modern-apis/

- History of APIs. (n.d.). API Evangelist History of APIs. Retrieved December 5, 2021, from https://history.apievangelist.com/

- What is API banking? Everything fintechs & banks need to know. (n.d.). Treasury Prime. Retrieved December 5, 2021, from https://www.treasuryprime.com/blog/api-banking

- 3 API Security Lessons from “Scorched Earth: Hacking Bank APIs.” (n.d.). Noname Secuirty. Retrieved December 5, 2021, from https://nonamesecurity.com/blog/3-api-security-lessons-from-scorched-earth-hacking-bank-apis

ABOUT KNIGHT INK

Firm Overview

Knight Ink is a content strategy, creation, and influencer marketing agency founded for category leaders and challenger brands in cybersecurity to fill current gaps in content and community management. We help vendors create and distribute their stories to the market in the form of written and visual storytelling drawn from 20+ years of experience working with global brands in cybersecurity. Knight Ink balances pragmatism with thought leadership and community management that amplifies a brand's reach, breeds customer delight and loyalty, and delivers creative experiences in written and visual content in cybersecurity.

Amid a sea of monotony, we help cybersecurity vendors unfurl, ascertain, and unfetter truly distinct positioning that drives accretive growth through amplified reach and customer loyalty using written and visual experiences.

Knight Ink delivers written and visual content through a blue ocean strategy tailored to specific brands. Whether it's a firewall, network threat analytics solutions, endpoint detection and response, or any other technology, every brand must swim out of a red sea of competition clawing at each other for market share using commoditized features. We help our clients navigate to blue ocean where the lowest price or most features don't matter.

We work with our customers to create a content strategy built around their blue ocean then perform the tactical steps necessary to execute on that strategy through the creation of written and visual content assets unique to the company and its story for the individual customer personas created in the strategy setting.

Contact Us

Web: www.knightinkmedia.com
Phone: (702) 637-8297
Address: 1980 Festival Plaza Drive, Suite 300, Las Vegas, NV 89135

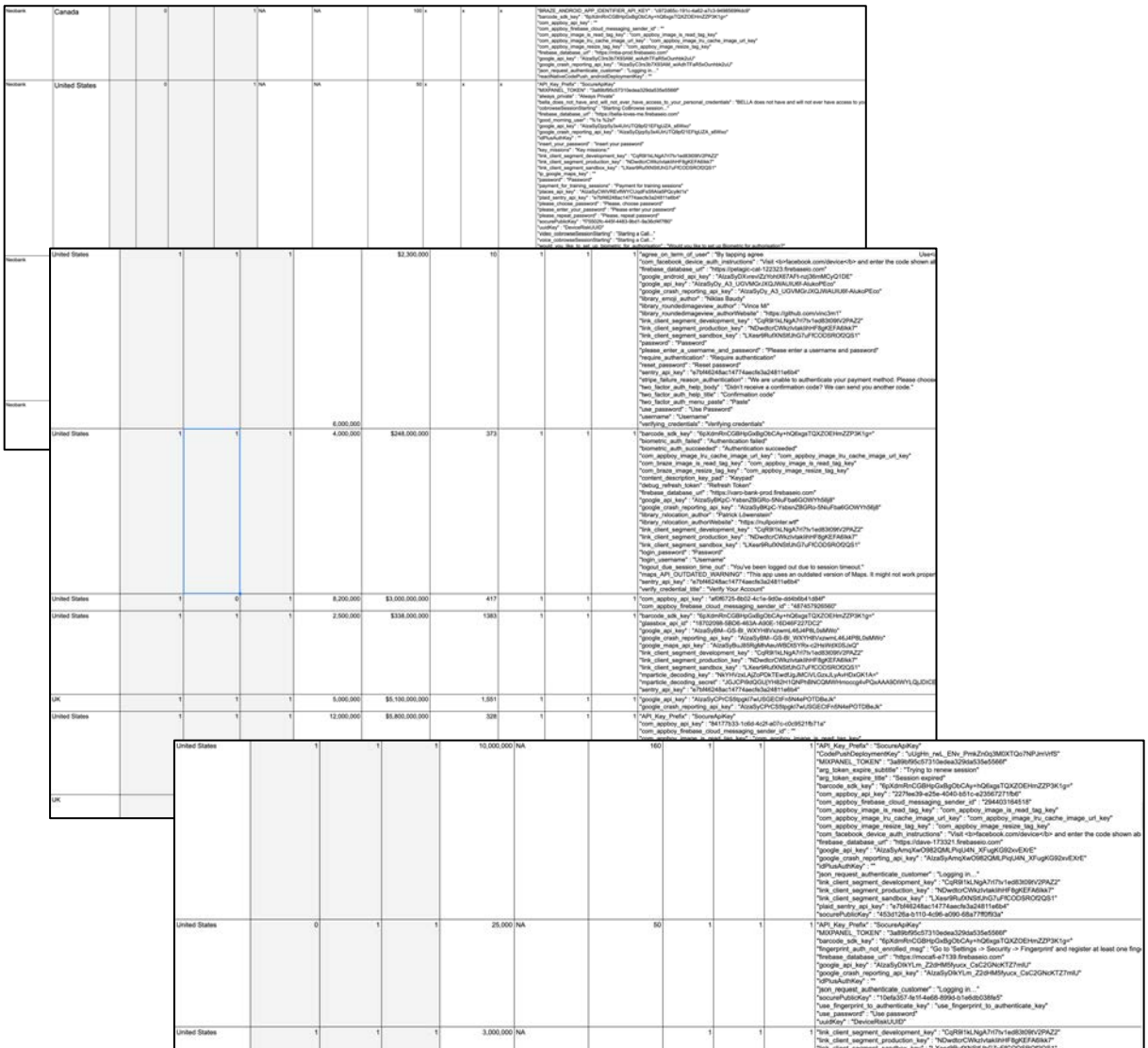
ABOUT NONAME SECURITY

Firm Overview

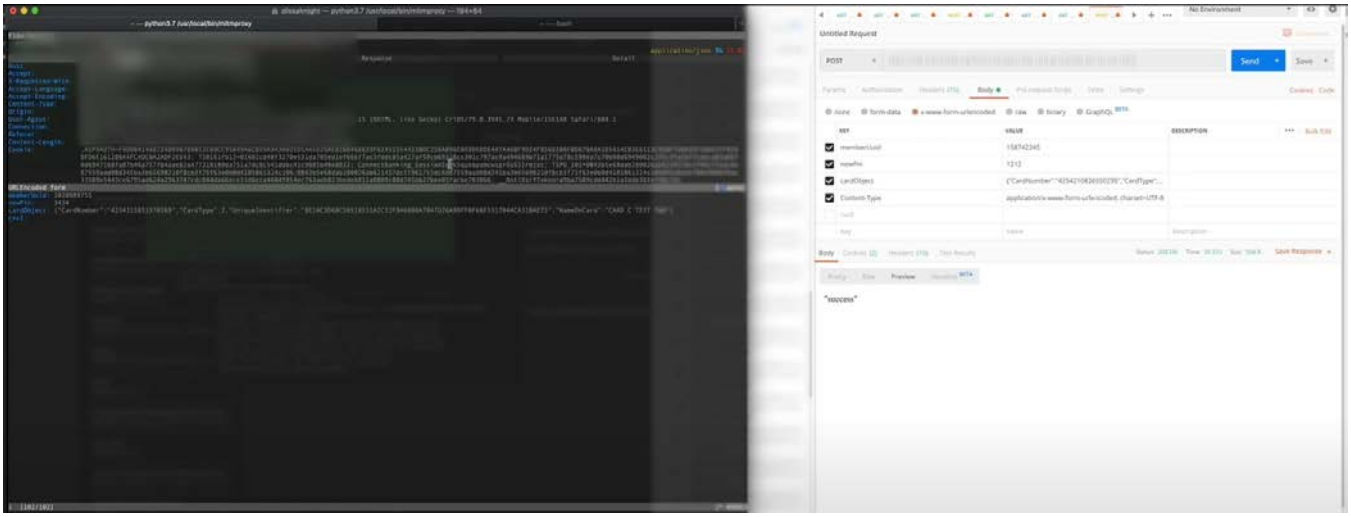
Noname Security is the only company taking a complete, proactive approach to API Security. Noname works with 20% of the Fortune 500 and covers the entire API security scope across three pillars — Posture Management, Runtime Security, and Secure API SDLC. Noname Security is privately held, remote first with headquarters in Palo Alto, California, and an office in Tel Aviv and London.

APPENDICES

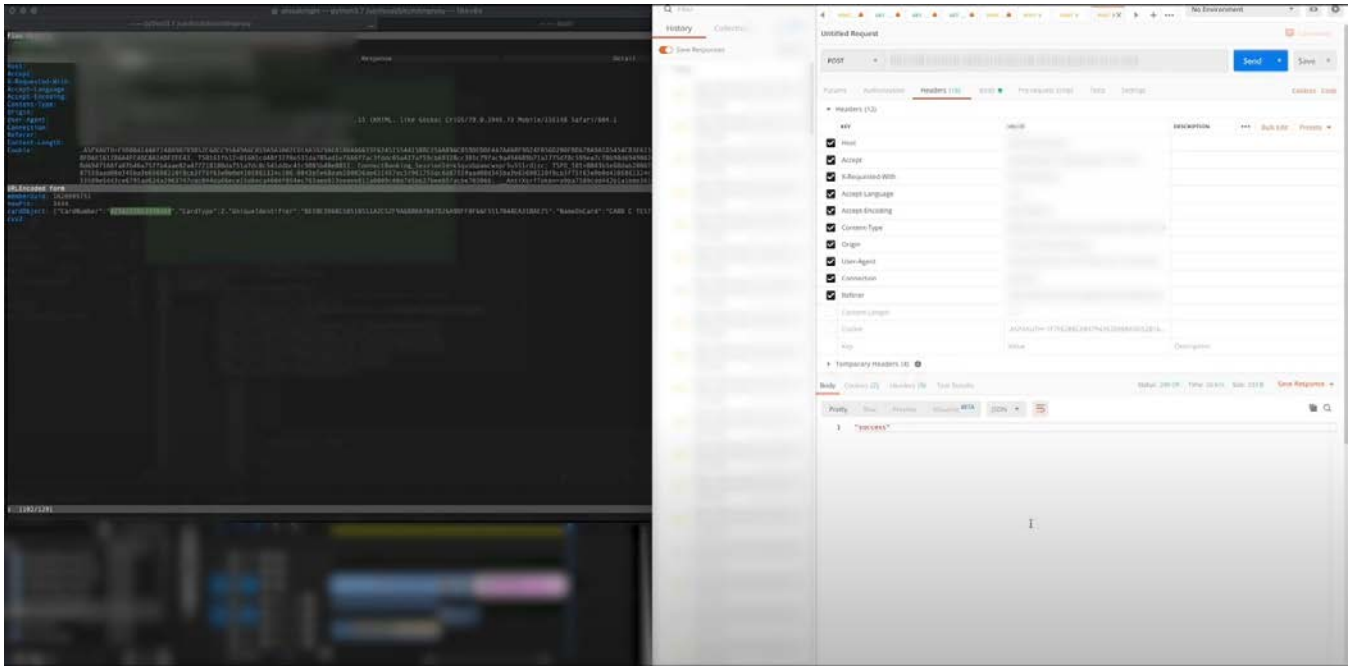
APPENDIX A



APPENDIX B



APPENDIX C


History
Untitled Request
POST
Send
Save
Headers (12)
Host
Accept
X-Requested-With
Accept-Language
Accept-Encoding
Content-Type
Origin
User-Agent
Connection
Referer
Content-Length
Cookie
Temporary Headers (8)
Body
Pretty
JSON
"success"

Knight Ink
1980 Festival Plaza Drive
Suite 300
Las Vegas, NV 89135
ak@knightinkmedia.com